Arbres de Décision Multi-Modes et Multi-Cibles

Frank Meyer, Fabrice Clerot

France Telecom R&D
Avenue Pierre Marzin
22307 Lannion cédex
franck.meyer@francetelecom.com
fabrice.clerot@francetelecom.com

Résumé. Nous présentons une nouvelle méthode d'induction d'arbre de décision appelée MuMTree (pour Multi Models Tree) utilisable pour les modes d'apprentissage supervisé, non supervisé, supervisé à plusieurs variables cibles. Nous présentons les différents principes nécessaires pour réaliser un tel arbre de décision. Nous illustrons ensuite, sur un cas de modélisation multi-cibles, les avantages de cette méthode par rapport à un arbre de décision classique.

1 Introduction

L'approche classique pour modéliser un problème avec N variables cibles est de décomposer le problème en N sous problèmes indépendants et d'utiliser un modèle par variable à expliquer. L'hypothèse sous jacente à cette décomposition est l'hypothèse d'indépendance des variables cibles entre elles. Dans de nombreux cas (données spatiales, données socio-économiques,...) cette hypothèse est fausse. Une méthode générant un seul modèle capable de prédire plusieurs variables cibles à la fois pourrait-elle être plus performante qu'une méthode avec plusieurs modèles spécialisés par variable cible ? Une telle méthode pourrait-elle être utilisée également pour l'apprentissage non supervisé ? Peut-on utiliser le cadre des arbres de décision, réputés pour leur efficacité et leur lisibilité pour construire une telle méthode ? Cet article apporte des réponses à ces différentes questions.

Il existe de très nombreuses méthodes d'induction d'arbre de décision. Les méthodes les plus connues sont pour le mode supervisé, (Kass, 1980), (Beiman, 1984), (Utgoff, 1991), (Quinlan, 1993), (Murthy, 1994) et pour le mode non supervisé, (Chavent, 1998), (Liu, 2000). A notre connaissance, aucune méthode d'induction d'arbres de décision ne permet de générer des modèles à la fois pour le mode d'apprentissage supervisé, non supervisé et multi-cibles.

L'objectif de MumTree est de fournir un principe général de construction d'arbres de décision pour les modes d'apprentissage supervisé et non supervisé, pour les types d'attributs courants (numériques, symboliques). Il en découlera aussi le mode supervisé multi-cibles.

2 Principes des Arbre multi-modes et multi-cibles

Nous passons en revue dans les sous-sections suivantes les 3 principes généraux d'une méthode d'induction d'arbres de décisions multi-modes et multi-cibles.

Notations générales

Soit A_1... A_m l'ensemble des attributs qui décrivent les instances.
Soit $x_1 \ldots x_n$ les instances proprement dites.
Soit Dom(A_1),.., Dom(A_m) les ensembles de valeurs prises par les attributs $A_1 ... A_m$.
On distinguera 2 cas :
- pour un attribut A_i numérique, Dom(A_i) est inclus dans $\mathbb{R}$, ensemble des réels
- pour un attribut A_i symbolique, alors Dom(A_i) est l'ensemble des valeurs symboliques (appelées modalités) de cet attribut.

2.1 Principe général du mode d'utilisation des attributs

Le mode d'utilisation des attributs dans les arbres de décisions classiques est toujours exclusif : un attribut est soit l'attribut cible, soit un attribut explicatif (s'il est ni l'un ni l'autre il est tout simplement ignoré). Dans le cas d'un arbre de décision multi-modes et multi-cibles, les modes d'utilisation des attributs ne doivent pas être exclusifs. Chaque attribut a deux propriétés propres, à valeurs booléennes, "cible" et "explicatif", correspondant à leur mode d'utilisation dans la génération du modèle. L'espace des attributs cibles sera celui sur lequel le critère d'évaluation des partitions sera calculé. L'espace des attributs explicatifs sera celui sur lequel les partitions seront effectuées. Dans le cas d'une modélisation supervisée à un attribut cible, l'attribut de classe est cible et non explicatif, les autres attributs étant explicatifs et non cibles ; ce principe est étendu à N attributs dans le cas d'une modélisation multi-cibles. Dans le cas d'une modélisation non-supervisée, tous les attributs sont à la fois cibles et explicatifs.

2.2 Principe d'évaluation des partitions

L'Inertie Intra-Classe (IIC) est un indicateur classique d'évaluation de la qualité d'une partition dans $\mathbb{R}^N$, et est donc un bon candidat comme critère d'évaluation. On doit cependant pouvoir calculer une IIC sur des données avec des attributs qui peuvent être aussi bien numériques avec des échelles différentes que symboliques. On doit donc utiliser un espace de recodage (noté Z) pour estimer l'inertie des données.

On choisira pour les attributs symboliques un codage disjonctif complet, et pour les attributs numériques un codage de type normalisation par l'étendue. Les attributs de recodage auront donc leurs valeurs dans [0;1] ou {0; ½ } suivant qu'ils sont issus d'attributs numériques ou symboliques. On doit normaliser les valeurs des attributs de recodage issus des attributs symboliques à ½ afin d'obtenir une distance maximale de 1 pour 2 valeurs différentes de l'attribut initial.

En sortie du recodage, tous les attributs recodés $Z_1 ... Z_Q$ sont numériques et ont tous [0;1] comme intervalle de valeurs. On fait donc correspondre à tout attribut numérique A_j son

attribut de recodage $Z_{j'}$ et à tout attribut symbolique A_j son ensemble d'attributs $Z_{j_1}...Z_{j_f}$ (f : nombre de modalités de A_j.). La fonction de distance $d_Z(.)$ pour les données dans le nouvel espace de représentation des attributs $Z_1...Z_Q$ est définie par :

$$d_Z(x,y) = \sqrt{\sum_{j=1}^{Q} \left((x'_j - y'_j)\ w_j \right)^2} \qquad (1)$$

avec (x,y) un couple d'instances issues de l'ensemble initial E, et (x',y') le couple issu du recodage dans Z du couple (x,y), et avec w_j= 1 si Z_j est issu du recodage d'un attribut A_i qui est cible, et w_j=0 sinon.

On peut utiliser toute fonction distance compatible avec une notion de centre de gravité dans un espace vectoriel. Nous avons choisi ici et dans la suite la distance euclidienne. L'Inertie d'un nuage X de n points, ayant pour centre de gravité g est :

$$I(X) = \frac{1}{n} \sum_{i=1}^{i=n} d\ (x_i, g)^2 \qquad (2)$$

avec d(.) : fonction de distance.

L'IIC de n points répartis dans une partition P composées de K sous-ensembles $C_1...C_K$ est définie par :

$$IIC(P) = \frac{1}{n} \sum_{j=1}^{K} \left(\sum_{x_i \in C_j} d^{\,2}(x_i, g_k) \right) \qquad (3)$$

avec : g_k : centre de gravité de l'ensemble des points appartenant au sous ensemble C_k, et avec d(.) : une fonction de distance (compatible avec la notion de barycentre).

2.3 Principe de croissance de l'arbre

Le critère utilisé pour évaluer les partitions sera donc l'Inertie Intra Classe (3) calculée en utilisant la fonction distance (1) sur l'espace de recodage des données Z défini précédemment. La croissance de l'arbre est guidée par la minimisation de l'IIC à chaque bipartition d'une feuille. Or l'IIC de clusters issus d'une partition n'est autre que la somme des inerties locales de chaque cluster. Le gain en Inertie $G(C_k)$ d'une bipartition (C_{k_1}, C_{k_2}) de C_k est défini par :

$$G(C_k) = I(C_k) \cdot |C_k| - I(C_{k_1}) \cdot |C_{k_1}| - I(C_{k_2}) \cdot |C_{k_2}| \quad (4)$$

avec la notation |E| : cardinal de l'ensemble E

Pour choisir la feuille à partitionner : on effectue pour chaque feuille C_k de l'arbre une recherche d'une bipartition minimisant l'IIC. Puis pour chacune des feuilles bipartitionnées on calcule le gain en inertie et on sélectionne la feuille qui maximise le gain en inertie.

2.4 Principes généraux d'exploration des bipartitions

Une règle à 1 ou plusieurs conditions binaires portant sur des attributs explicatifs permet de définir une bipartition d'un ensemble de données E en 2 sous ensembles P_1 et P_2. Chaque condition est par exemple du type $A_i < V_{ij}$ dans le cas d'un attribut A_i numérique, ou du type $A_i=M_{ij}$ dans le cas d'un attribut symbolique.

Nous avons étudié plusieurs façons de générer les règles de décision de l'arbre : des méthodes polythétiques permettant de créer des arbres avec des règles à plusieurs conditions à chaque test, et des méthodes monothétiques générant des règles avec une seule condition. Les détails de cette expérimentation et ses résultats, non précisés ici par manque de place, sont disponibles sur simple demande aux auteurs. Nous avons retenu par la suite une méthode monothétique gloutonne, reproductible et d'un niveau de performance correct : les attributs numériques sont discrétisés en K intervalles, par une discrétisation simple (equals width). Les modalités peu fréquentes des attributs symboliques sont regroupées ; la recherche du meilleur point de coupure est effectuée sur les attributs explicatifs sur les intervalles de valeurs et les groupes de valeurs. On sélectionne l'attribut, le test (= ou <> pour un attribut symbolique, <= ou > pour un attribut numérique) et la valeur qui minimisent l'IIC de la bipartition

3 Cas d'utilisation

3.1 Note sur la comparaison théorique entre MumTree et CART

Notons que dans le cas d'une analyse supervisée à 2 classes, le critère de l'inertie intra-classe sur l'attribut cible recodé par codage disjonctif complet est identique au critère de Gini à un facteur multiplicatif près : un arbre MumTree a donc un comportement de croissance similaire à un arbre CART, à condition d'adopter le même mode d'arrêt.

Actuellement le critère d'arrêt implémenté dans MumTree est la profondeur maximale de l'arbre à générer. On pourrait par la suite implémenter un post-pruning comme dans CART.

3.2 Illustration de l'utilisation dans le cas multi-cibles

Nous avons comparé, sur des données synthétiques, les performances de MumTree avec CART dans un cas d'apprentissage supervisé à 2 variables cibles.

L'ensemble de données utilisé pour le test comprend 2 variables explicatives et 2 variables cibles et représente une fonction de $\mathbb{R}^2$ dans $\mathbb{R}^2$ transformant des coordonnées polaires en coordonnées cartésiennes. Les variables explicatives sont l'angle et le rayon, les variables cibles sont les coordonnées cartésiennes X et Y. L'ensemble de données est composé de 1000 instances de 4 attributs. L'angle varie de 0 à 2π selon une distribution uniforme. Le rayon varie de 0.8 à 1 selon une distribution uniforme.

L'utilisation de CART pour réaliser un modèle multi-cibles nécessite soit de passer par une discrétisation du plan, soit d'utiliser 2 modèles de régression (un pour X, un pour Y).

Cas de la discrétisation : Nous avons discrétisé l'espace de sortie en 10×10 cellules. CART prend comme variable cible le résultat de la discrétisation (100 cellules, donc 100 valeurs différentes). CART utilise les variables explicatives angle et longueur, et comme variable cible les cellules du plan discrétisé. MumTree utilise les mêmes variables explicatives, un intervalle de discrétisation de 10 et les variables de sorties X et Y comme cibles. Les résultats des prédictions obtenus en TEST sont donnés dans les Figures 1 et 2. On voit que CART a du mal à reconstituer la couronne. Cela est dû au nombre important de modalités (100) de la variable cible. Mais diminuer le nombre de modalités reviendrait à diminuer la résolution de discrétisation, avec des résultats moins bons. MumTree reconstruit lui 10 groupes répartis autour de la couronne, qui apparaît mieux reconstituée.

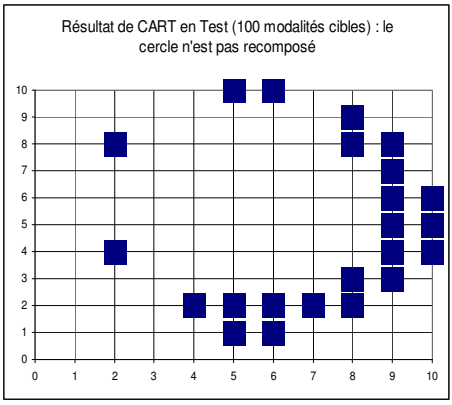

FIG. 1 – *Résultat de CART en TEST, cas d'une variable cible multi-modale issue d'une discrétisation du plan en 10×10 intervalles.*

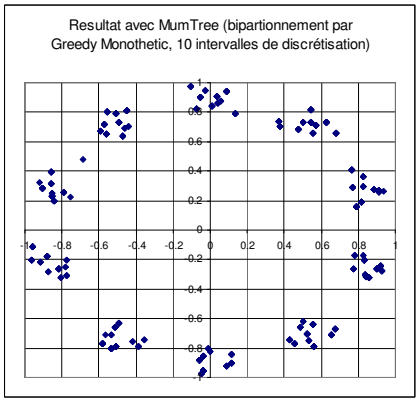

FIG. 2 – *Résultat de MumTree en Test, 2 variables cibles, Greedy Monothetic avec 10 intervalles.*

Cas de 2 modèles pour CART : dans ce cas nous avons modélisé avec CART la variable X et la variable Y séparément. CART est utilisé cette fois-ci en tant que modèle de régression. Avec MumTree, nous avons réalisé un seul modèle, mais en prenant cette fois-ci un nombre d'intervalles de discrétisation non limité, égal au nombre d'instances en apprentissage. Les Figures 3 et 4 montrent respectivement les résultats obtenus pour CART et MumTree. Dans ce cas, MumTree reconstitue légèrement mieux la couronne initiale (avec 1 seul modèle).

4 Conclusion

Nous avons montré qu'il était possible de généraliser le fonctionnement d'un arbre de décision pour qu'il puisse travailler indépendamment en mode supervisé, non supervisé et multi-cibles. Nous avons détaillé les principes permettant de réaliser un tel arbre de décision : architecture générale, gestion des attributs, protocole de recodage des données dans $\mathbb{R}^N$, choix d'un critère normalisé (l'inertie) à minimiser dans l'espace des attributs cibles.

Nous avons montré sur un cas simple le fonctionnement de l'arbre de décision MumTree démontrant l'apport du mode multi-cibles notamment comparativement à une méthode classique mono-cible (CART).

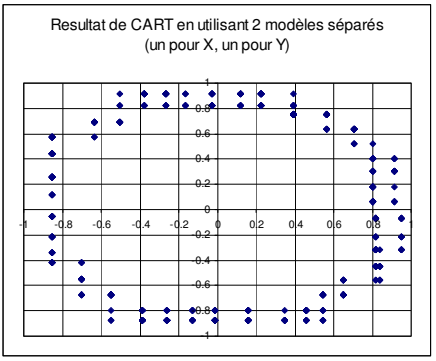

FIG. 3 – *Résultat de CART en TEST, cas de 2 modèles de régression.*

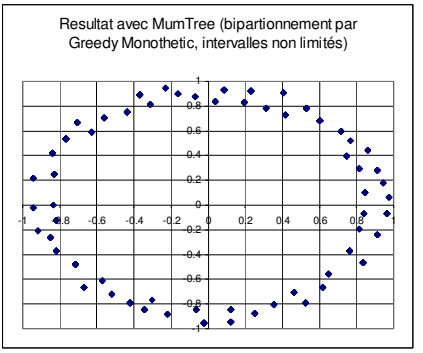

FIG. 4 – *Résultat de MumTree en Test, 2 variables cibles, Greegy Monothetic avec intervalles=nombre d'instances.*

Références

Breiman, L and J.H. Friedman, R.A. OLSHEN and C.J. STONE (1984), "Classification and Regression Trees", Chapman et Hall.

Chavent, M. (1998), "A monothetic clustering method", *Pattern Recognition Letters* 19, 989-996.

Kass, G.V., (1980), An Exploratory Technique for Investigating Large Quantities of Categorical Data. Applied Statistics 29: p. 119-127.

Liu, B., Y. Xia, and P. Yu, (2000), "Clustering through decision tree construction", *Technical Report RC21695,* IBM Research.

Murthy, Sreerama K, Simon Kasif, and Steven Salzberg. (1994) A System for Induction of Oblique Decision Trees. Journal of Artificial Intelligence Research, 2:1–32.

Quinlan, J.R. (1996), "Induction of decision trees", *Machine Learning 1*, 86-106.

Utgoff, P.E. and C.E. Brodley, (1990): An Incremental Method for Finding Multivariate Splits for Decision Trees. Proc. of the 7th International Conference on Machine Learning.

Summary

This paper introduces a new Decision Tree method called MumTree (Multi Models Tree). MumTree can be used in the supervised learning mode, in the unsupervised learning mode, and in the supervised learning mode with multi-targets model. This paper describes the general principles used in this Decision Tree method. Finally, we illustrate the advantages of our method compared to a classical decision tree (CART) on a multi-targets modelling example.